

## 2005/01/20

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office



【特許請求の範囲】

✓ 【請求項1】複数の処理モジュールからなる一連の処理を実行するための情報処理システムであって、前記複数の処理モジュールの少なくとも一部を実行するハードウェア・モジュールを構成するとともに、ハードウェア・モジュールの部分的書き換えが可能なプログラマブル論理回路と、

前記プログラマブル論理回路上に構成するハードウェア・モジュールを記憶する記憶手段であって、同一機能を実行し、入出力部位置が異なる複数のハードウェア・モジュールをハードウェア・モジュール群として記憶する記憶手段と、

前記プログラマブル論理回路に構成する新規構成ハードウェア・モジュールを、前記記憶手段中のハードウェア・モジュール群から選択して構成するハードウェア・モジュール取得手段と、

を有することを特徴とする情報処理システム。

✓ 【請求項2】前記ハードウェア・モジュール取得手段は、前記プログラマブル論理回路上の他のハードウェア・モジュールとの入出力部位置との配線距離が最短となる入出力部位置を有するハードウェア・モジュールを新規構成ハードウェア・モジュールとして前記ハードウェア・モジュール群から選択する構成を有することを特徴とする請求項1に記載の情報処理システム。

【請求項3】前記情報処理システムは、各処理モジュールがプログラム言語で記述されたソフトウェア・モジュールで構成されるプログラムの形態で一連の処理を受容するプログラム受容手段と、ソフトウェア・モジュールを実行可能なプロセッサと、回路情報で記述されたハードウェア・モジュールに従って部分書き換え可能なプログラマブル論理回路と、一連の処理を構成する各処理モジュールのうちハードウェア・モジュールによって実行すべきものを決定する実行モジュール決定手段とを有し、

前記記憶手段は、ソフトウェア・モジュールと同じ処理を前記プログラマブル論理回路上で再構成するための回路情報で記述されるハードウェア・モジュールを蓄積する構成を有し、

前記ハードウェア・モジュール取得手段は、前記実行モジュール決定手段によって決定されたハードウェア・モジュールを前記記憶手段から取り出して前記プログラマブル論理回路上に再構成する構成を有することを特徴とする請求項1または2に記載の情報処理システム。

✓ 【請求項4】前記情報処理システムは、前記プログラマブル論理回路上に構成したハードウェア・モジュールの識別符号を記録する識別符号記録手段を有し、

前記ハードウェア・モジュール取得手段は、前記識別符号記録手段に記録された識別符号に基づいて、前記プログラマブル論理回路上の他のハードウェア・モジュール

の入出力部位置を識別する構成を有することを特徴とする請求項1乃至3いずれかに記載の情報処理システム。

✓ 【請求項5】前記記憶手段に蓄積されるハードウェア・モジュールは矩形形状で表現されることを特徴とする請求項1乃至4いずれかに記載の情報処理システム。

✓ 【請求項6】ハードウェア構成可能領域の形状は前記プログラマブル論理回路上における水平方向と垂直方向の各々の論理セル数によって定義されることを特徴とする請求項1乃至5いずれかに記載の情報処理システム。

【請求項7】前記記憶手段に記憶されるハードウェアモジュールの入出力部位置は、前記記憶手段に蓄積される矩形形状のハードウェア・モジュールの外周位置に設定された構成であることを特徴とする請求項1乃至6いずれかに記載の情報処理システム。

【請求項8】複数の処理モジュールからなる一連の処理を実行する情報処理方法であって、前記複数の処理モジュールの少なくとも一部を実行するハードウェア・モジュールを記憶手段から選択するモジュール選択ステップと、

前記モジュール選択ステップにおいて選択されたハードウェア・モジュールをプログラマブル論理回路に構成するモジュール構成ステップと、を有し、

前記モジュール選択ステップは、同一機能を実行し入出力部位置が異なる複数のハードウェア・モジュールから、前記プログラマブル論理回路上の他のハードウェア・モジュールとの入出力部位置との配線距離が最短となる入出力部位置を有する1つのハードウェア・モジュールを選択するステップを含むことを特徴とする情報処理方法。

【請求項9】前記情報処理方法は、さらに、一連の処理を構成する各処理モジュールのうち、プロセッサを使用してソフトウェア・モジュールによって実行すべきものと、前記プログラマブル論理回路を使用してハードウェア・モジュールによって実行すべきものを決定する実行モジュール決定ステップを有し、

前記モジュール選択ステップは、前記実行モジュール決定ステップにおいてハードウェア・モジュールによって実行すべきものとして決定されたハードウェア・モジュールについて前記記憶手段から選択することを特徴とする請求項8に記載の情報処理方法。

【請求項10】前記モジュール選択ステップは、前記プログラマブル論理回路上の他のハードウェア・モジュールの識別符号から一義的に求められる回路構成に基づいて、該他のハードウェア・モジュールとの入出力部位置を判別して、該他のハードウェア・モジュールとの配線距離が最短となるハードウェア・モジュールを新規構成ハードウェア・モジュールとして選択することを特徴とする請求項8または9に記載の情報処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、論理回路を記述した回路情報に従って所望の論理を再構成可能なプログラマブル論理回路を含んだ情報処理システムに係り、特に、プログラム言語で記述された複数の処理モジュールで構成される一連の処理の少なくとも一部を、部分書き換え可能なプログラマブル論理回路上で再構成したハードウェア回路に置き換えて実行するタイプの情報処理システムに関する。

【0002】更に詳しくは、本発明は、各処理モジュールをソフトウェア・モジュール又はハードウェア・モジュールのいずれで実行すべきかを決定して処理するタイプの情報処理システムに係り、プログラム論理回路上に形成された異なるモジュール間での信号伝達時間、いわゆる配線遅延を減少して処理モジュール全体の処理時間を短縮化する情報処理システムに関する。

【0003】

【従来の技術】昨今の技術革新に伴い、各種の情報処理機器が研究・開発され、広汎に普及してきている。情報処理機器は、一般に、高度な半導体技術によって集積化された半導体回路コンポーネント (LSI: Large Scale Integration) を数多く含んで構成される。LSIの一部には、特定の用途のために最適設計されたカスタムIC、すなわちASIC (Application Specific Integrated Circuit) 技術が採用されている。

【0004】最近のASIC分野では、製品の開発期間を更に短縮化するために、フィールド・プログラマブル・ゲートアレイ (FPGA: Field Programmable Gate Array) や、プログラマブル・ロジック・デバイス (PLD: Programmable Logic Device)、CPLD (Complex Programmable Logic Device) などの、プログラマブル論理回路装置が広く使われ始めている。

【0005】この種のプログラマブル論理回路装置は、論理回路を記述する回路情報を該論理回路装置内に読み込ませることで、内部の論理回路と論理回路間の結線を自由に構成することができる。これによって、従来は回路設計の終了後に数週間から数か月を必要とした集積回路の作製時間を短縮することができる。特に、米国特許第4,700,187号明細書に開示されているプログラマブル論理装置は、電気的に再構成が可能な構成であり、一度作製した回路を必要に応じて自由に何度でも変更する、すなわち回路を再構成できるという利点があり、ますます広く使われるようになってきている。

【0006】FPGAと呼ばれるプログラマブル論理回路装置は、論理ゲート間および配線間を接続するスイッチを制御するビット・ストリームを外部から書き込むことによって、所望の論理回路を再構成することのできるデバイスである。内部回路構成や論理ゲート間のスイッ

チ制御のデータを保持する記憶手段がデバイス内に設けられており、この記憶手段の種類により様々な動作特性のデバイスが当業界において提供されている。

【0007】論理ゲート間または配線間のスイッチ制御のデータを保持する記憶手段としては、アンチヒューズ・タイプ、EEPROM (Electrically Erasable and Programmable Read Only Memory) タイプ、SRAM (Static Random Access Memory) タイプなどがある。

【0008】このうち、SRAMタイプは、スイッチとなるトランジスタのゲートに揮発性メモリを接続する。書き換え可能デバイス中でSRAMメモリの占める面積は大きい、他のタイプに比べ書き換え時間が小さく、アドレスを指定してメモリの内容を書き換えることができるので、システム動作中の論理の再構成も可能である。さらに、再構成されない部分は動作を継続するという機能を実現し、デバイス内の部分書き換えを可能にしている。

【0009】リアルタイムに部分書き換えが可能なFPGAは、特に、DR-FPGA (Dynamically Reconfigurable Field Programmable Gate Array) と呼ばれている。DR-FPGAとしては、米国ザイリンクス社のXC6200シリーズ及びVirtexシリーズ、米国アトメル社のAT6000シリーズ及びAT40Kシリーズ、米国ルーセント・テクノロジー社のORCA3C/3Tシリーズなどが、既に市販されている。

【0010】最近の論理回路に要求される処理の複雑さが増し、単一のプログラマブル論理回路装置だけでは実現できない規模にまで回路規模が大きくなっている。この問題を解決するための1つの方法として、処理の途中でプログラマブル論理回路装置を再度構成し直すことによって、異なる時間に異なる論理回路を実現することが考えられている。

【0011】この方法によれば、モバイル機器のような極度に小型化された情報処理システムのように、内蔵可能な回路規模に制約がある場合でも、回路コンポーネント数を増やすことなく、様々な処理をプログラマブル論理回路によってハードウェア化して比較的高速に実行できるという利点もある。

【0012】しかしながら、プログラマブル論理装置を再構成するときに、回路情報を再度読み込ませるための余分な時間がかかるという欠点がある。また、処理の途中で論理装置を再構成するためには、処理を一時中断し、その時点でのデータをプログラマブル論理回路装置の外部の記憶装置に一時記憶させ、新たな回路情報を読み込んで再構成し、再構成前のデータと再構成に伴う新しいデータを入力させる、という一連の余分な処理が必要になる。

【0013】プログラマブル論理装置の再構成に伴う上記の問題を解決するために、米国アトメル社が提供するデータブック"CONFIGURABLE LOGIC"に記載されているプログラマブル論理回路装置、及び、米国ザイリンクス社が提供するデータブック"THE PROGRAMMABLE LOGIC"（米国版）に記載されているプログラマブル論理回路装置では、再構成を行うときにデータを記憶するためのデータ記憶装置を有しており、回路の動作中でも外部の記憶装置から回路情報の一部を読み込んで部分的に再構成を行うことで、論理装置を再構成するための時間を最小限に留めている。

【0014】ところが、このような部分的に再構成可能なプログラマブル論理回路を情報処理システムに好適に適用するには、情報処理システム上で実行する処理を、プログラマブル論理回路で行うハードウェア処理と、マイクロプロセッサなどの回路構成を変更できない固定的な論理回路装置で行うソフトウェア処理とに分離するための技術を確立する必要がある。

【0015】このような処理の分離に関する問題を解決する方法として、プログラマブル論理回路の再構成機能を用いてアプリケーションの一部をハードウェア化する手法が幾つか提案されている。アプリケーションの一部を、汎用のプロセッサによるソフトウェア処理ではなく、プログラム論理回路装置を用いたハードウェア処理に委ねるのは、後者の方が前者よりも処理速度が速いという理由に依拠する。

【0016】例えば、特開平9-74556号公報には、ネットワークを介してアプリケーションの一部をダウンロードする手法について記載している。すなわち、同公報では、プログラマブル論理回路を用いたアプリケーションとして、符号化した画像データと復号プログラムをネットワーク経由で受信し、ソフトウェアとプログラマブル論理回路の双方を用いて復号化して画像再生するということが考えられている。図11には、同公報に開示されている画像再生装置の概略構成を示している。以下、図11を参照しながら、この画像再生装置について説明する。

【0017】画像再生装置は、動画像復号部を再構成するためのプログラムと符号化された動画像データをネットワーク経由でダウンロードするための受信部と、受信データがプログラムか動画像データかによって伝送先を切り換える切り換え手段と、処理系を再構成することが可能な動画像復号部と、受信したプログラムを動画像復号部の構成に変更できる形式に変換するプログラム変換部と、受信部と切り換え手段とプログラム変換部と動画像復号部の制御を行うための制御部とで構成される。

【0018】プログラム変換部は、受信プログラムを動画像復号部の構成を変更できる形式に変換する第1プログラム部分（すなわち、ハードウェア化する部分）と動

画像再生時において制御部が行う第2プログラム部分（すなわち、ソフトウェアで実行する部分）とに分割する分割部と、第1プログラム部分を動画像復号部の構成を変更できる形式に変換する第1の変換部（ハードウェア・コンパイラ）と、第2プログラム部分を制御部が解読して処理を実行できる形式に変換する第2の変換部（ソフトウェア・コンパイラ）とを備えている。

【0019】動画像復号部は、逆量子化処理内容を変更できる逆量子化部と、逆変換処理内容を変更できる逆変換部とを備えている。逆量子化部は量子化された画像データを逆量子化し、逆変換部は変換された画像データを逆変換する。

【0020】次に、この画像再生装置の動作手順について説明する。

【0021】動画像再生用のプログラムと符号化された動画像データがネットワーク経由で入力されると、まず、論理記述言語で記述された再生アルゴリズム・プログラムを受信部で受信する。

【0022】受信データはプログラム変換部に伝送され、分割部でハードウェア処理部とソフトウェア処理部とに分割される。

【0023】ハードウェア処理部分は、第1の変換部に渡され、プログラマブル論理回路装置（FPGA）を書き換えることができる形式（ビット・ストリーム）に変換され、制御部は、このビット・ストリームに基づいて、逆量子化部と逆変換部を書き換える。

【0024】他方、ソフトウェア処理部分は、第2の変換部に渡され、プロセッサで処理できる実行形式に変換された後、制御部に渡される。

【0025】こうして再構成された画像再生装置に対して、動画像データがネットワーク経由で送られると、受信部で受信し、切り換え手段によって動画像復号部に伝送され、復号データとして出力する。

【0026】上述したように逆量子化部と逆変換部をFPGAのような処理プログラムを変更可能な素子で構成したことにより、動画像復号部のハードウェアを最適な処理能力に適合させるようにしている訳である。

【0027】また、特開平6-301522号には、回路構成を変更できない固定部とプログラマブル論理回路のように回路構成を変更できる可変部とで構成される計算機のシステム構成方法について開示している。同公報は、複数のプログラムを基に、プログラマブル論理回路上にハードウェアで構成することを提案するものである。以下、図12を用いながら、この計算機システムについて説明する。

【0028】図12に示すように、計算機で実行するソースプログラムは、固定部の構成に関する情報と可変部が構成することができる回路の情報が格納されているライブラリを参照しながら、オブジェクト・コードとハードウェア構成データに変換される。

【0029】すなわち、コンパイラはソース・プログラムのフロー解析を行い、関数の頻度を検出する。この検出頻度に基づいて、呼び出し回数の多い関数をハードウェアで処理する関数として決定し、ハードウェア構成データを作成して出力する。次に、ハードウェアで処理すると決めた部分を所定の可変部で処理することを示すコードを残りのソフトウェアで処理する部分に付加して、オブジェクト・コードを作成し出力する。

【0030】計算機システムは、回路構成を変更できない固定部と、ハードウェア構成データにより構成された可変部を用いて、オブジェクト・コードに応じた処理を実行する。このようにして、コンパイル時に呼び出し回数の多い幾つかの関数をハードウェア化することにより処理全体の高速化を図っている。

【0031】また、日経オープンシステム1997年8月号の213ページから219ページに掲載されている記事「インターネットのサーバを運用代行 ISPやVAN業者がサービスを提供へ」では、ネットワークを介して複数のサーバにアクセスするケースについて提案している。

【0032】同記事は、ネットワークのアウトソーシングというサービスとしてインターネットやクローズドVANについて扱っている。この中で、インターネットに不可欠なサーバであるメール・サーバやネーム・サーバを始めとする各種サーバへのアクセスについて詳解している。

【0033】サーバへのより早いアクセスを実行するために、使用するプログラム一つ一つについてアクセスを行なうよりも、全てのプログラムについて一度にアクセスを行い、最も早くアクセスできたプログラムからダウンロードする方が、全体のアクセス時間が少ないことが当業界において既に知られている。そこで、プログラムが複数のサーバに格納されているような状況では、上記のような方法に従ってプログラムへのアクセスを行い、ダウンロードすることが一般的になってきている。例えば、プログラム・データを分類、整理したり、管轄が異なっている、あるいはネットワークのトラフィック緩和、サーバの負荷軽減のために、プログラムは複数のサーバに分散して蓄積される。

【0034】しかしながら、上記で説明した各々の従来例では、処理の一部分をプログラマブル論理回路で処理するに際して、次のような問題が残されている。

【0035】すなわち、プログラマブル論理回路を用いた処理における問題点として、以前より、ハードウェア・モジュールをデバイス上に構成した場合における様々な状況での配線遅延が指摘されている。

【0036】プログラマブル論理回路において行われる論理回路の部分書き換えによるハードウェア・モジュールの再構成処理において、再構成されたハードウェア・モジュール間の接続を行う場合、従来の装置では、ハ

ードウェア・モジュールのインターフェース位置はモジュールが提供された時点で固定されている。従って、接続すべき二つのハードウェア・モジュールの接続部が隣接した位置にあるか、あるいはそれぞれのハードウェア・モジュールと対称の位置、すなわち離れた位置に接続インターフェースがあるかによって、ハードウェア・モジュール接続部で発生する配線遅延時間が変化してしまう。

【0037】その結果、二つのハードウェア・モジュール、例えばモジュールAとモジュールBとの相互接続部がそれぞれのハードウェア・モジュールと対称の位置にある場合には、モジュールAの処理後、モジュールBでの処理を開始するために実行されるモジュールAの処理情報のモジュールBに対する伝達時間に多大な時間を要することになる。すなわち配線遅延が発生する。そのために、ハードウェアで処理を行うことによる処理の高速化というメリットを削減してしまうという問題を発生させていた。また繰り返し処理がこの複数のハードウェア・モジュールを使用したルーチン処理として行われる場合、例えばモジュールAとモジュールBとの処理を繰り返し実行する必要がある場合は、配線遅延が繰り返し処理の行われる回数に応じて加算されてしまい、処理時間がさらに増加してしまうという問題を引き起こしていた。

【0038】

【発明が解決しようとする課題】本発明は、上述の問題点に鑑みてなされたものであり、本発明の目的は、論理回路を記述した回路情報に従って所望の論理を再構成可能なプログラマブル論理回路を含んだタイプの、優れた情報処理システムを提供することにある。

【0039】本発明の更なる目的は、プログラム言語で処理を記述された複数の処理モジュールで構成される一連の処理の少なくとも一部を部分書き換え可能なプログラマブル論理回路上で再構成したハードウェア回路に置き換えて実行するタイプの、優れた情報処理システムを提供することにある。

【0040】本発明の更なる目的は、各処理モジュールをソフトウェア・モジュール又はハードウェア・モジュールのいずれで実行すべきかを決定して処理するタイプの、優れた情報処理システムを提供することにある。

【0041】本発明の更なる目的は、従来のプログラマブル論理回路において問題となっていたハードウェア・モジュール間の配線遅延を減少させる構成を実現し、再構成可能なプログラマブル論理回路を使用した情報処理における処理時間の短縮を可能とした情報処理システムを提供することを目的とする。

【0042】

【課題を解決するための手段】本発明は、上記課題を参酌してなされたものであり、その第1の側面は、複数の処理モジュールからなる一連の処理を実行するための情

報処理システムであって、前記複数の処理モジュールの少なくとも一部を実行するハードウェア・モジュールを構成するとともに、ハードウェア・モジュールの部分的書き換えが可能なプログラマブル論理回路と、前記プログラマブル論理回路上に構成するハードウェア・モジュールを記憶する記憶手段であって、同一機能を実行し、入出力部位置が異なる複数のハードウェア・モジュールをハードウェア・モジュール群として記憶する記憶手段と、前記プログラマブル論理回路に構成する新規構成ハードウェア・モジュールを、前記記憶手段中のハードウェア・モジュール群から選択して構成するハードウェア・モジュール取得手段とを有することを特徴とする情報処理システムにある。

【0043】さらに、本発明の情報処理システムにおいて、前記ハードウェア・モジュール取得手段は、前記プログラマブル論理回路上の他のハードウェア・モジュールとの入出力部位置との配線距離が最短となる入出力部位置を有するハードウェア・モジュールを新規構成ハードウェア・モジュールとして前記ハードウェア・モジュール群から選択する構成を有することを特徴とする。

【0044】さらに、本発明の情報処理システムは、各処理モジュールがプログラム言語で記述されたソフトウェア・モジュールで構成されるプログラムの形態で一連の処理を受容するプログラム受容手段と、ソフトウェア・モジュールを実行可能なプロセッサと、回路情報で記述されたハードウェア・モジュールに従って部分書き換え可能なプログラマブル論理回路と、一連の処理を構成する各処理モジュールのうちハードウェア・モジュールによって実行すべきものを決定する実行モジュール決定手段とを有し、前記記憶手段は、ソフトウェア・モジュールと同じ処理を前記プログラマブル論理回路上で再構成するための回路情報で記述されるハードウェア・モジュールを蓄積する構成を有し、前記ハードウェア・モジュール取得手段は、前記実行モジュール決定手段によって決定されたハードウェア・モジュールを前記記憶手段から取り出して前記プログラマブル論理回路上に再構成する構成を有することを特徴とする。

【0045】さらに、本発明の情報処理システムは、前記プログラマブル論理回路上に構成したハードウェア・モジュールの識別符号を記録する識別符号記録手段を有し、前記ハードウェア・モジュール取得手段は、前記識別符号記録手段に記録された識別符号に基づいて、前記プログラマブル論理回路上の他のハードウェア・モジュールの入出力部位置を識別する構成を有することを特徴とする。

【0046】さらに、本発明の情報処理システムにおいて、前記記憶手段に蓄積されるハードウェア・モジュールは矩形形状で表現されることを特徴とする。

【0047】さらに、本発明の情報処理システムにおいて、ハードウェア構成可能領域の形状は前記プログラマ

ブル論理回路上における水平方向と垂直方向の各々の論理セル数によって定義されることを特徴とする。

【0048】さらに、本発明の情報処理システムにおいて、前記記憶手段に記憶されるハードウェアモジュールの入出力部位置は、前記記憶手段に蓄積される矩形形状のハードウェア・モジュールの外周位置に設定された構成であることを特徴とする。

【0049】さらに、本発明の情報処理方法は、複数の処理モジュールからなる一連の処理を実行する情報処理方法であって、前記複数の処理モジュールの少なくとも一部を実行するハードウェア・モジュールを記憶手段から選択するモジュール選択ステップと、前記モジュール選択ステップにおいて選択されたハードウェア・モジュールをプログラマブル論理回路に構成するモジュール構成ステップと、を有し、前記モジュール選択ステップは、同一機能を実行し入出力部位置が異なる複数のハードウェア・モジュールから、前記プログラマブル論理回路上の他のハードウェア・モジュールとの入出力部位置との配線距離が最短となる入出力部位置を有する1つのハードウェア・モジュールを選択するステップを含むことを特徴とする。

【0050】さらに、本発明の情報処理方法は、さらに、一連の処理を構成する各処理モジュールのうち、プロセッサを使用してソフトウェア・モジュールによって実行すべきものと、前記プログラマブル論理回路を使用してハードウェア・モジュールによって実行すべきものを決定する実行モジュール決定ステップを有し、前記モジュール選択ステップは、前記実行モジュール決定ステップにおいてハードウェア・モジュールによって実行すべきものとして決定されたハードウェア・モジュールについて前記記憶手段から選択することを特徴とする。

【0051】さらに、本発明の情報処理方法は、前記モジュール選択ステップは、前記プログラマブル論理回路上の他のハードウェア・モジュールの識別符号から一義的に求められる回路構成に基づいて、該他のハードウェア・モジュールとの入出力部位置を判別して、該他のハードウェア・モジュールとの配線距離が最短となるハードウェア・モジュールを新規構成ハードウェア・モジュールとして選択することを特徴とする。

【0052】

【作用】本発明に係る情報処理システムは、複数の処理モジュールを用いて一連の処理を実行する。各処理モジュールは基本的にプログラム言語で記述されたソフトウェア・モジュールで構成されており、情報処理システムは、所謂ソフトウェア・プログラムの形態で、一連の処理を受容する。

【0053】また、情報処理システムは、ソフトウェア・モジュールを実行可能なプロセッサと、回路情報で記述されたハードウェア・モジュールに従って部分書き換え可能なプログラマブル論理回路を備えている。一連の



処理を実行する際には、各処理モジュールをソフトウェア・モジュール又はハードウェア・モジュールのいずれとして実行すべきかを判別して、実行することが可能である。

【0054】また、情報処理システムは、ソフトウェア・モジュールと同じ処理を前記プログラマブル論理回路上で再構成するための回路情報で記述されるハードウェア・モジュールを蓄積する記憶手段を備えている。

【0055】ハードウェア・モジュールは、プログラマブル論理回路上に再構成すると、所定数のセルを使用するとともに、所定形状を持つセル領域を占有する。本発明では、各ソフトウェア・モジュールにつき、同一の処理を実現するが前記プログラマブル論理回路上に再構成すると、その入出力部位置が異なる複数のハードウェア・モジュールを、記憶手段に用意した。

【0056】そして、プログラマブル論理回路の部分的書き換えを行なうときには、同じ処理を実現するがその入出力部位置が異なる複数のハードウェア・モジュールのうち、配線遅延を最小にするものを優先的に取り出すようにした。

【0057】本発明によれば、同じ処理を実現するがその入出力部位置が異なるハードウェア・モジュールを複数個用意しておき、プログラマブル論理回路において、配線遅延を最小にするハードウェア・モジュールを優先的に使用することにしたので、モジュール間でのデータ入出力時の遅延を最小限に抑えることができる。また、その結果として全体の処理時間を短縮化することができる。

【0058】また、ハードウェア・モジュールの入出力部の位置を、占有するセル領域を包含する矩形の外周上に配置することとした。すなわち、入出力部位置が異なる同機能のハードウェア・モジュールのパターン数を限定して、かつ最適な入出力部位置を有するモジュールを選択可能として、最適構成を少量のデータから判断することができ、処理の実行とデバイスのコンフィギュレーションを並行して行なうことができ、全体の処理時間を短縮することができる。

【0059】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【0060】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【0061】図1には、本発明を実現するのに適した情報処理システム10のハードウェア構成を模式的に示している。

【0062】情報処理システム10は、例えば、「ワークステーション」や「パーソナル・コンピュータ」と呼ばれる汎用コンピュータ・システムであり、その一例は、米国IBM社のPC/AT互換機又はその後継機で

ある。この種のシステム10は、OAG (PC Open Architecture Developer's Group) 仕様に準拠し、オペレーティング・システム(OS)として、例えば米国マイクロソフト社の"Windows 95/98/NT"を搭載している。以下、情報処理システム10内の各部について説明する。

【0063】システム10のメイン・コントローラであるCPU (Central Processing Unit) 11は、オペレーティング・システム(OS)の制御下で、各種の処理を実行する。

【0064】ここで言う「処理」は、複数の処理モジュールからなる一連の処理である。かかる一連の処理は、通常、各処理モジュールがプログラム言語で記述されたソフトウェア・モジュールで構成されている「プログラム」の形態で、情報処理システム10に供給される。

【0065】CPU 11の回路論理は固定的で、再構成不可能である。したがって、CPU 11は、処理モジュールのうち、プログラム言語で記述されたソフトウェア・モジュールのみを実行し、後述するようにハードウェア・モジュールは実行しない。なお、CPU 11は、例えば米国インテル社の"Pentium II"でよい。

【0066】本実施例の情報処理システム10は、各周辺装置をローカル接続するためのローカル・バスとしてのPCI (Peripheral Component Interconnect) バス14を備えている。このPCIバスと、CPU 11の自身の外部ピンに直結したホスト・バス11Bとは、チップセットで構成されるバス・ブリッジ13を介して相互接続されている。

【0067】本実施例のチップセット13は、両バス11B及び14間のデータ転送速度の差を吸収するためのデータ・バッファや、主記憶メモリ12へのアクセス動作を制御するためのメモリ・コントローラを含んだ構成となっている。

【0068】主記憶メモリ12は、書き込み可能なメモリであり、通常は複数個のDRAM (Dynamic Random Access Memory) チップで構成される。主記憶メモリ12は、BIOS (Basic Input/Output System: 基本入出力システム)、デバイス・ドライバ、OS、及びアプリケーション・プログラムなど各種処理を実現するためのプログラム・コードをロードしたり、処理実行中の作業データを一時格納するために利用される。

【0069】PCIバス14は、比較的高速なデータ転送が可能なバスであり(バス幅32/64ビット、最大動作周波数33/66MHz、最大データ転送速度132/264Mbps)、PCIバス14とのインターフェース・プロトコルを実現する各種のインターフェース・アダプタが接続されている。なお、PCIアーキテク

チャは、米国インテル社の提唱に端を発したものであり、所謂PnP（プラグ・アンド・プレイ）機能を備えている。

【0070】PCIバス14に接続されるインターフェース・アダプタとして、本実施例すなわち図1では、ハード・ディスク・インターフェース17と、通信インターフェース19と、プログラマブル論理回路インターフェース15を挙げている。

【0071】ハード・ディスク・インターフェース17は、ハード・ディスク装置18などの大容量の外部記憶装置を接続するためのインターフェースであり、IDE（Integrated Drive Electronics）やSCSI（Small Computer System Interface）などの仕様が当業界において標準的である。以下では、システム10内のハード・ディスク装置18のことを、「ローカル・ディスク」とも呼ぶ。

【0072】通信インターフェース19は、LAN（Local Area Network）やインターネットなどの、システム10の外部のネットワーク20と接続するためのアダプタである。

【0073】ネットワーク20上には、通常、複数の外部装置が接続されている。外部装置の一例は、ファイル・サーバやプリント・サーバなど、各種資源サービスを情報処理システム10に提供するサーバである。図1では、ネットワーク20には記憶装置21が接続されている。この記憶装置21は、例えば、ファイル・サーバなどが提供する大容量ハード・ディスク装置などであり、システム10側からは記憶装置21に対して透過的にディスク・アクセス可能である。以下では、ネットワーク20上のハード・ディスク装置21のことを、「ローカル・ディスク」と区別して、「ネットワーク・ディスク」と呼ぶことにする。

【0074】プログラマブル論理回路インターフェース15は、プログラマブル論理回路16をシステム10に接続するためのアダプタである。プログラマブル論理回路16は、例えば「アダプタ・カード」に搭載された形態で提供される。プログラマブル論理回路16は、後述するように部分書き換え動作が可能なタイプのデバイスである。プログラマブル論理回路16上に部分的に書き込まれたハードウェアのことを、以下では「ハードウェア・モジュール」と呼ぶ。ハードウェア・モジュールは、通常、論理回路16を再構成するための回路情報という形式で記述され、所定の格納場所に保管され、且つ適宜取り出して用いられる。

【0075】本実施例では、回路情報として記述された数多くのハードウェア・モジュールが、情報処理システム10内のローカル・ディスク18又はネットワーク20上のネットワーク・ディスク21に蓄積されている。プログラマブル論理回路16は、図示しないローカル・

メモリを備えている。部分書き換えを行なうときには、必要なハードウェア・モジュールを、ローカル・ディスク18又はネットワーク・ディスク21からローカル・メモリに一旦ダウンロードして、書き換え処理を行う。論理回路16の部分書き換え動作の手順については、後に詳解する。

【0076】なお、情報処理システム10を構成するためには、図1に示した以外にも多くのハードウェア構成要素等が必要である。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【0077】次に、プログラマブル論理回路16の内部構造について説明する。

【0078】図2には、プログラマブル論理回路16の構造を論理的に図解している。同図に示すように、プログラマブル論理回路16は、回路情報を格納するためのコンフィギュレーション・メモリ160と、回路素子164とで構成される。回路素子は、後述するように、論理セル161と、配線領域164と入出力端子163を含んだ集合体である。

【0079】また、図3には、プログラマブル論理回路16を構成するチップ内の物理構造を模式的に示している。同図に示すように、略矩形形状のチップ内16には、数多の論理セル161がマトリックス状に配設されている。配線領域162は、各論理セル161間を走るように、格子状に敷設されている。チップの周辺部には、論理回路16に対する電気信号データの入出力を行なうための数多の入出力端子163が列設されている。

【0080】コンフィギュレーション・メモリ160は、論理セル161内及び配線領域162内の回路情報を書き込むための記憶手段であり、SRAM、DRAMなどの書き換え可能なメモリ素子で構成されている。

【0081】コンフィギュレーション・メモリ160にはアドレスが割り振られており、あるアドレスに新しい回路情報を記述したデータが格納されると、アドレスに該当する論理セル161内の回路構成と、論理セル161及び入出力端子163を相互に接続する配線領域162の接続状態が、この新しい回路情報に従って再構成される。

【0082】図2に示すように、プログラマブル論理回路16に再構成されて形成された回路素子に処理すべきデータが入力されると、またその処理を実行した結果が出力されるようになっている。

【0083】プログラマブル論理回路16の回路構成を再構成する一連の動作のことを、「コンフィギュレーション」と呼ぶ。本実施例に係るプログラマブル論理回路16では、コンフィギュレーション・メモリ160の一部分のみを書き換えることで、プログラマブル論理回路

16が動作中であっても、回路を部分的に再構成することができる。すなわち、部分書き換え機能を提供している。

【0084】図4には、情報処理システム10が複数の処理モジュールからなる一連の処理を実行する仕組みを模式的に示している。但し、一連の処理は、プログラム言語で記述されたプログラム、すなわちアプリケーション・プログラム100の形態で情報処理システム10に供給されている（例えば、ハード・ディスク装置18内に導入されている）ものとする。図4に記載のプログラム100は、各処理モジュールがプログラム言語で記述されたソフトウェア・モジュールで構成されるプログラム形態で一連の処理を受容するプログラム受容手段である。

【0085】アプリケーション・プログラム100は、実行する一連の処理を複数の処理に分割可能であり、分割された各処理毎に1つのモジュールを形成している。このモジュールのことを、本明細書中では「処理モジュール」と呼んでいる。

【0086】アプリケーション・プログラム100を構成する各処理モジュールは、原初的には、CPU11において処理が実行可能な形式、すなわち、プログラム言語で記述されたソフトウェア・プログラムである。処理モジュール単位のソフトウェア・プログラムのことを、本明細書では「ソフトウェア・モジュール」と呼ぶ。言い換えれば、アプリケーション・プログラム100は、複数のソフトウェア・モジュールの集合体であると言える。

【0087】各ソフトウェア・モジュールが行なうのと同じ処理を、プログラマブル論理回路16上に構成したハードウェアとして実行することもできる。プログラマブル論理回路16によるハードウェア処理は、一般に、対応するソフトウェア処理よりも高速である。

【0088】ソフトウェア・モジュールに対応したハードウェアをプログラム論理回路16上に再構成するための回路情報を記述した単位モジュールのことを、本明細書では「ハードウェア・モジュール」と呼んでいる。図4中の記憶装置200には、アプリケーション・プログラム100を構成する各ソフトウェア・モジュールと同じ処理を実現するハードウェア・モジュールが蓄積されている。但し、記憶装置200の実体は、図1中で示したローカル・ディスク18やネットワーク・ディスク21であると把握されたい。

【0089】ここで、アプリケーション・プログラム100の構造についてもう少し詳しく言及する。

【0090】図5には、アプリケーション・プログラム100の構成を模式的に示している。同図に示すように、アプリケーション・プログラム100は、ヘッダ部と本体部からなる。

【0091】本体部は、前述したように、複数のソフ

トウェア・モジュールの集合で構成されている。ヘッダ部には、各ソフトウェア・モジュールに対応するハードウェア・モジュールの各々についての識別符号が記述されている。

【0092】本実施例では、ハードウェア・モジュールは、対応するソフトウェア・モジュールと同じ処理をプログラマブル論理回路16上に再構成するための回路情報だけでなく、その回路情報をプログラマブル論理回路16上に再構成したときに占有するセル領域の形状を記述している。ここで、ハードウェア・モジュールの形状は、占有するセル領域を完全に包含する矩形として表現される。したがって、形状は、その矩形が持つ水平及び垂直の各方向のPFU (Programmable Function Unit) 数すなわちセル数によって定義されるので、少量のデータで済む。

【0093】各ハードウェア・モジュールの識別番号は、同じ処理を行うソフトウェア・モジュールとの対応関係が明瞭となるように、付与されることが好ましい。図4及び図5に示した例では、アプリケーション・プログラム100上の処理実行順に従った順序番号によって、各ソフトウェア・モジュールに対応するハードウェア・モジュールの識別符号がヘッダ部に記述される。このような場合、繰り返し使用する処理モジュールに対応するハードウェア・モジュールの識別番号は、ヘッダ部において繰り返し記述されることになる。

【0094】また、本実施例の記憶装置200に記憶されたハードウェア・モジュールは、同一機能を実行するがその入出力部位置を異なる形態とした複数のモジュール（例えばn個（nは1以上の整数））を1つの集合としたモジュール群として記憶している。図4に記載の例では、ハードウェア・モジュール群1〜6とあるように、例えばハードウェア・モジュール群1としてk種類、ハードウェア・モジュール群2にm種類等、それぞれ同一機能を実行するがその入出力部位置を異なる形態とした複数のモジュールが記憶されている。

【0095】これらの同一機能を実行するがその入出力部位置を異なる形態とした複数のモジュールは、再構成可能なハードウェアであるプログラマブル論理回路400に同時期に形成される他のモジュールとの情報入出力用配線を最短とするモジュールを、これら複数のモジュール中から選択可能とするために用意されたものであり、この構成の詳細については後述する。

【0096】なお、必ずしも各ハードウェア・モジュールについて、記憶装置200中に複数種類登録されていることは必須ではなく、例えば再構成可能なハードウェアであるプログラマブル論理回路全体を使用して構成されるハードウェア・モジュールであれば、プログラマブル論理回路に同時に形成される他のモジュールとの情報伝達の必要がないので、1つのモジュールパターンが形成できればよい。

【0097】再び図4に戻って、情報処理システム10上でアプリケーション・プログラム100を実行する仕組みについての説明を継続する。

【0098】ハードウェア・モジュール取得手段300と、実行モジュール決定手段600は、処理モジュールをハードウェア・モジュールとして実行する動作に深く関与している。これらの手段300及び600は、例えば、情報処理システム10上に導入されたOSの機能の一部、という形態で実装することができる。

【0099】ハードウェア・モジュール取得手段300は、アプリケーション・プログラム100のヘッダ部に記述されている識別符号を用いて、ソフトウェア・モジュールが実行する処理と同じ処理を実行するハードウェア・モジュールを記憶装置200から取得する。そして、ハードウェア・モジュール取得手段300は、取得したハードウェア・モジュールを実現した回路をプログラマブル論理回路16上に再構成する。

【0100】図4に示すように、ハードウェア・モジュール取得手段300は、取得要求送出手段310と、取得再構成手段320とで構成される。ハードウェア・モジュールを記録装置200から取得するとき、取得要求送出手段310は、必要な全てのハードウェア・モジュールの取得要求を行う。

【0101】実行モジュールをCPU11によるソフトウェア・モジュールとするか、又は、プログラマブル論理回路16によるハードウェア・モジュールとするかを、プログラムの実行時に決定することもできる。この場合には、取得要求送出手段310は、アプリケーション・プログラム100のヘッダ部に記述されている全てのハードウェア・モジュールの識別符号を含むハードウェア・モジュール取得要求を送出する。

【0102】また、実行モジュールの決定をプログラム100の実行前に行うこともできる。この場合には、取得要求送出手段310は、ハードウェア・モジュール処理により実行すると決定されたハードウェア・モジュールの識別符号を含むハードウェア・モジュール取得要求を送出する。

【0103】ハードウェア・モジュール取得手段300内の取得再構成手段320は、記憶手段200から取得したハードウェア・モジュールの情報を、それに付加されている識別符号により認識し、システム10内に未だ取り込んでいないハードウェア・モジュールであることを確認した後に、プログラマブル論理回路16上にそのハードウェア・モジュールを実現した回路を再構成する。

【0104】このとき、プログラマブル論理回路16上に新たに形成するハードウェア・モジュール（例えばモジュールC）として、同一機能を実行するがその入出力部位置を異なる形態とした複数のモジュール（C-a～h）が記憶装置200に登録されている場合は、プログ

ラムブル論理回路16上にすでに形成された別のハードウェア・モジュール（例えばモジュールA、B）との情報入出力位置関係に応じて、入出力部配線距離が最短のものをモジュール群（C-a～h）の中から選択する。また、再構成したハードウェア・モジュールの識別符号（例えばC-c）を識別符号記録手段800に記録するとともに、システム10内の記憶装置200（例えば、ハード・ディスク装置18や主記憶メモリ12）に、そのハードウェア・モジュールの情報を格納する。これらのモジュール選択処理についてはさらに後述する。

【0105】他方、アプリケーション・プログラム100のソフトウェア・モジュールによるソフトウェア処理は、CPU11が実行する。

【0106】実際のアプリケーション・プログラム100の処理においては、実行モジュール決定手段600が、ハードウェア又はソフトウェアのどちらのモジュールで行うかを各処理モジュール毎に決定し、実行する。

【0107】図4では示していないが、実行モジュール決定手段600に選択条件設定手段を設けてもよい。選択条件設定手段は、ソフトウェア・モジュール及びハードウェア・モジュールの各々による処理時間、メモリ消費量、プログラマブル論理回路16の再構成時間などの種々の選択条件項目を予め想定しておき、その1つあるいは複数の条件項目の組み合わせからなる実行モジュール選択条件を設定する。実行モジュール決定手段600は、選択条件を評価することによって、実行モジュールを適切に決定することができる。

【0108】図6は本実施例をより具体的に説明するための図である。本実施例では、処理A、処理B、処理CをA、B、Cの順に2回繰り返す、すなわち、A→B→C→A→B→Cのシーケンスで、プログラマブル論理回路上へハードウェア・モジュールをコンフィギュレーションし、それぞれの処理をコンフィギュレーションされた回路上で順に行っていくことを考える。

【0109】図6のハードウェア・モジュール（A）、（B）、（C）の黒い四角はそれぞれ、各モジュールの入出力部を示している。ここでは簡単のため、それぞれのモジュールの形状を矩形形状とし、また各モジュールは他のモジュールに対して一つの入出力部を持つものとして話を進める。

【0110】各ハードウェア・モジュールが格納されている記憶部には、図6の下部に示すように、処理Cに必要なハードウェア・モジュールのうち、機能が同じで入出力部の位置が異なっている（C-a）、（C-b）、（C-c）、（C-d）、（C-e）、（C-f）、（C-g）、（C-h）の複数のモジュールデータがハードウェア・モジュール群として格納されている。

【0111】図6に示すようにこの中でハードウェア・モジュール（A）及びハードウェア・モジュール（B）の入出力部と最も最短距離で接続可能なハードウェア・

モジュールは(C-c)であり、構成モジュールとしてモジュール(C-c)が選択される。

【0112】次に、図6に示すような処理モジュールA、B、Cから成るハードウェア・モジュールによって処理を実行する手順について詳解する。図7には、この手順をフローチャートの形式で示している。以下、このフローチャートの各ステップについて説明する。

【0113】但し、処理が実行されるアプリケーション・プログラム100は、ハードウェア・モジュールとして実行される3つの処理モジュールA、B、及びCを含み、且つ、A→B→C→A→B→Cという順番で各処理モジュールが実行されることとする。

【0114】既に述べたように、ハードウェア・モジュール取得手段300と、実行モジュール決定手段600は、情報処理システム10上で動作するOSにおける各1つの機能、すなわちソフトウェア的に実装されている。したがって、この動作は、アプリケーション・プログラム100の開始にตอบสนองして、OSによって処理が実行される。

【0115】アプリケーション・プログラム100が開始されると、プログラム100内のヘッダ部に記述されているハードウェア・モジュールの識別符号(図5を参照のこと)が読み込まれる(ステップS11)。既に述べたように、識別符号は、各ソフトウェア・モジュールと同じ処理をプログラマブル論理回路16上に再構成する回路情報と、再構成されたハードウェア・モジュールの形状に関する情報を含んでいる。

【0116】ここで、ハードウェア・モジュールの形状は、ハードウェア・モジュールをプログラマブル論理回路16上に再構成して占有するセル領域を完全に包含する矩形で表現することとし、したがって、矩形が持つ水平及び垂直の各方向のセル(PFU)数によって定義されている。

【0117】予定された順序に従って各処理モジュールを逐次実行するという形式で、アプリケーション・プログラム100の実行が進行する。そして、ハードウェア・モジュールによる処理部分に到達すると、必要なハードウェア・モジュールがプログラマブル論理回路16上にコンフィギュレーションされていく。図7の破線で囲んだ部分がこれに相当する。

【0118】本実施例では、アプリケーション・プログラム100は、ハードウェア・モジュールとして実行すべきと決定されている処理モジュールとして、A、B、Cという3つを含み、且つ、処理モジュールの実行順序はA→B→C→A→B→Cと指定されている。

【0119】まず、ステップS12において、処理Aを実現するハードウェア・モジュール(A)が、プログラマブル論理回路16上にコンフィギュレーションされる。次いで、ステップS13において、処理Aが開始される。

【0120】上述のステップS12における処理Aのモジュールのコンフィギュレーションの終了と同時にハードウェア取得手段300は、記憶手段200から、次の処理Bに必要なハードウェア・モジュール(B)の取得を試みる。処理Bに必要なハードウェア・モジュール(B)がプログラマブル論理回路16上にコンフィギュレーションされる(S23)。処理Aが終了すると、ハードウェア・モジュール(B)のコンフィギュレーションが終了したかどうかを確認して(S14)、終了している場合は処理Bを実行する(S15)。

【0121】ハードウェア・モジュール(B)のコンフィギュレーションが終了すると同時にハードウェア取得手段300は、記憶手段200から、次の処理Cに必要なハードウェア・モジュール(C)の取得を試みて、処理Cに必要なハードウェア・モジュール(C)をコンフィギュレーションする。ここで、各ハードウェア・モジュールが格納されている記憶手段200には、処理Cに必要なハードウェア・モジュールとして、機能が同じで入出力部の位置が異なっている(C-a)、(C-b)、(C-c)、(C-d)、(C-e)、(C-f)、(C-g)、(C-h)の複数種類を格納している。

【0122】ハードウェア・モジュール取得手段300は、このモジュール群の中でハードウェア・モジュール(A)及びハードウェア・モジュール(B)の入出力部と最も最短距離で接続可能なハードウェア・モジュールを選択する(S24)。この時点で、すでにプログラム論理回路16上に形成されているハードウェア・モジュール(A、B)の識別符号に含まれる回路情報に基づいて、記憶装置200中のモジュールCに関するモジュール群、すなわち機能が同じで入出力部の位置が異なっている(C-a)、(C-b)、(C-c)、(C-d)、(C-e)、(C-f)、(C-g)、(C-h)から、ハードウェア・モジュール(A)及びハードウェア・モジュール(B)の入出力部と最も最短距離で接続可能なハードウェア・モジュールを選択する。このモジュールは図6で示すように(C-c)であるので、このモジュール(C-c)を構成モジュールとして選択する。

【0123】次に、ステップS25において、選択されたハードウェア・モジュール(C-c)をプログラマブル論理回路16上にコンフィギュレーションする。処理Bが終了すると、処理Cに必要なハードウェア・モジュール(C-c)がコンフィギュレーションされたかどうかを確認して(S16)、コンフィギュレーションされていれば処理Cを実行する(S17)。処理Cが終了すると、この処理が1回目の処理であるかどうかを確認し(S18)、1回目であれば、処理Aの実行(S13)に戻り、そうでなければハードウェア・モジュールでの処理を終了する。

【0124】図8(a)及び(b)に、本実施例及び従

来例におけるハードウェア・モジュール(C)をプログラマブル論理回路上にコンフィギュレーションした場合の入出力部の接続状況の比較図を概念的に示す。本発明のコンフィギュレーション例が(a)図であり、従来のコンフィギュレーション例が(b)図である。ここでは、従来例では、図6に示すハードウェア・モジュール(C-g)のパターンのみが構成可能であると想定する。

【0125】図8(a)におけるハードウェア・モジュール(C-c)とハードウェア・モジュール(A)及びハードウェア・モジュール(B)との間の入出力部を接続する配線群と比較して、図8(b)におけるハードウェア・モジュール(C-g)とハードウェア・モジュール(A)及びハードウェア・モジュール(B)との間の入出力部を接続する配線群は、ハードウェア・モジュール(C-g)の入出力部の位置がプログラマブル論理回路上でハードウェア・モジュール(A)及びハードウェア・モジュール(B)の入出力部位置近傍の辺上に配置されていないため、配線距離が長くなってしまい、これに伴う配線遅延が生じてしまう。

【0126】図8に示す状態で、本実施例及び従来例におけるハードウェア・モジュール(C)をプログラマブル論理回路上にコンフィギュレーションした場合の入出力部の配線遅延について図9を用いて説明する。

【0127】図9には、本実施例に従って処理A→処理B→処理C→処理A→処理B→処理Cの順で各ハードウェア・モジュールが実行される処理フローを、従来例と比較しながら、時系列的に表示している。すなわち、同図の(a)が本実施例であり、同図(b)が従来例である。

【0128】図9において、 $t_A$ 、 $t_B$ 、 $t_C$ はハードウェア・モジュール(A)、ハードウェア・モジュール(B)、ハードウェア・モジュール(C)のプログラマブル論理回路上のコンフィギュレーション時間を示す。 $T_A$ 、 $T_B$ 、 $T_C$ は処理A、処理B、処理Cの処理時間を示す。 $T_{AB}$ 、 $T_{BC}$ 、 $T_{CA}$ はそれぞれハードウェア・モジュール(A)からハードウェア・モジュール(B)、ハードウェア・モジュール(B)からハードウェア・モジュール(C)、ハードウェア・モジュール(C)からハードウェア・モジュール(A)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達時間を示す。

【0129】以下、時系列に従って図9について説明する。

【0130】まず、本実施例における処理フローについて言及する。時刻 $t_0$ において、処理Aに必要なハードウェア・モジュール(A)のコンフィギュレーションが開始する。次いで、時刻 $t_1$ において、ハードウェア・モジュール(A)のコンフィギュレーションが終了すると、処理Bに必要なハードウェア・モジュール(B)の

コンフィギュレーションが開始される。また、これと同時に、ハードウェア・モジュール(A)によって処理Aが開始される。

【0131】時刻 $t_2$ において、ハードウェア・モジュール(B)のコンフィギュレーションが終了すると、処理Cに必要なハードウェア・モジュールのうち、ハードウェア・モジュール群からすでに構成済みのハードウェア・モジュール(A)とハードウェア・モジュール

(B)に対して、配線遅延が最小となる構成、すなわちハードウェア・モジュール(A)とハードウェア・モジュール(B)の入出力部と最短距離で接続可能なモジュール(C-c)を選択的に取得して、コンフィギュレーションを開始する。

【0132】時刻 $t_3$ において、処理Aが終了すると、ハードウェア・モジュール(A)からハードウェア・モジュール(B)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、 $T_{AB}$ である。ハードウェア・モジュール(A)からハードウェア・モジュール(B)へのデータ入力終了すると、時刻 $t_4$ において、ハードウェア・モジュール(B)を用いて処理Bが開始する。時刻 $t_5$ は、モジュール(C-c)のコンフィギュレーション終了時刻である。

【0133】さらに、プログラマブル論理回路16上にコンフィギュレーションされたハードウェア・モジュール(B)を用いた処理Bの終了とともに、ハードウェア・モジュール(B)からハードウェア・モジュール(C-c)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、 $T_{BC}$ である。本発明の構成では、ハードウェア・モジュール(B)からハードウェア・モジュール(C-c)へは最短距離の入出力配線構成が実現されているので、この信号伝達に要する時間： $T_{BC}$ は非常に短時間で終了する。

【0134】さらに、プログラマブル論理回路16上にコンフィギュレーションされたハードウェア・モジュール(C-c)を用いた処理Cの終了とともに、ハードウェア・モジュール(C-c)からハードウェア・モジュール(A)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、 $T_{CA}$ である。本発明の構成では、ハードウェア・モジュール(C)からハードウェア・モジュール(A)へは最短距離の入出力配線構成が実現されているので、この信号伝達に要する時間： $T_{CA}$ は短時間で終了する。

【0135】以降、同様の処理A、B、Cが各処理間にデータ入出力処理、 $T_{AB}$ 、 $T_{BC}$ を介して実行される。

【0136】次に、従来例における処理フローについて言及する。時刻 $t_0$ において、処理Aに必要なハードウ

ウェア・モジュール(A)のコンフィギュレーションを開始する。次いで、時刻 $t_1$ において、ハードウェア・モジュール(A)のコンフィギュレーションが終了すると、処理Bに必要なハードウェア・モジュール(B)のコンフィギュレーションを開始される。同時にハードウェア・モジュール(A)によって処理Aが開始される。

【0137】時刻 $t_2$ において、ハードウェア・モジュール(B)のコンフィギュレーションが終了すると、処理Bの次に実行される処理Cに必要なハードウェア・モジュールをプログラマブル論理回路16上にコンフィギュレーションする。しかしながら、従来例では、処理Cを実行するハードウェア・モジュールとして(C-g)のみしか記憶装置には記憶されておらず、図8(b)に示す入出力部位置を持つハードウェア・モジュール(C-g)が構成されることになる。

【0138】時刻 $t_3$ において、処理Aが終了すると、ハードウェア・モジュール(A)からハードウェア・モジュール(B)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、TABである。ハードウェア・モジュール(A)からハードウェア・モジュール(B)へのデータ入力が終了すると、時刻 $t_4$ において、ハードウェア・モジュール(B)を用いて処理Bが開始する。時刻 $t_5$ は、モジュール(C-g)のコンフィギュレーション終了時刻である。

【0139】さらに、プログラマブル論理回路16上にコンフィギュレーションされたハードウェア・モジュール(B)を用いた処理Bの終了とともに、ハードウェア・モジュール(B)からハードウェア・モジュール(C-g)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、TBCである。従来例の構成では、ハードウェア・モジュール(B)からハードウェア・モジュール(C-g)へは図8(b)から理解されるように、本発明の構成(図8(a))に比較して長い入出力配線構成となっているので、この信号伝達に要する時間：TBCは本発明の構成(モジュール(C-c)を構成した場合)に比較して長い時間となる。

【0140】さらに、プログラマブル論理回路16上にコンフィギュレーションされたハードウェア・モジュール(C-g)を用いた処理Cの終了とともに、ハードウェア・モジュール(C-g)からハードウェア・モジュール(A)へ、各モジュールの入出力部間を接続する配線群を介した信号伝達が行われる。この信号伝達に要する時間は、TCAである。従来例の構成では、ハードウェア・モジュール(C-g)からハードウェア・モジュール(A)へは図8(b)から理解されるように、本発明の構成(図8(a))に比較して長い入出力配線構成となっているので、この信号伝達に要する時間：TCAは本発明の構成に比較して長い時間を要することにな

る。

【0141】以降、同様の処理A、B、Cが各処理間にデータ入出力処理、TAB、TBCを介して実行される。

【0142】アプリケーションが開始されてから一連の処理が終了するまでの流れは図7のフローチャートを用いて前述した通りであるが、ここで、ハードウェア・モジュール(B)とハードウェア・モジュール(C)、及びハードウェア・モジュール(C)とハードウェア・モジュール(A)の間の本実施例と従来例の配線遅延差を $T'BC$ 、 $T'CA$ とすると、それぞれ、

$$T'BC = t_8 - t_7,$$

$$T'CA = (t_{12} - t_{11}) - (t_{10} - t_9)$$

である。

【0143】従って、 $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$ のシーケンスで処理を行うと、本発明では従来例より配線遅延によって生じる全体の処理時間増大を $(2T'BC + T'CA)$ 減少させることが可能となる。

【0144】本実施例では処理シーケンスの繰り返し数を2回としたが、従来例において配線遅延が生じる部分の繰り返し処理数が多くなればなるほど、本発明の効果は大きくなる。例えば、本実施例において繰り返し数を2回ではなく、M回(M：自然数)とすると、その効果として $(MT'BC + (M-1)T'CA)$ だけ配線遅延によって生じる全体の処理時間増大を減少させることが可能となる。

【0145】なお、本実施例では簡単のため、各ハードウェア・モジュールを矩形形状として考えたが、実際には様々な形状が考えられるものであり、コンフィギュレーションされるハードウェア・モジュールの外枠の形状に応じて入出力部を配置することが可能であることはいうまでもない。

【0146】また、本実施例では入出力部を一つのハードウェア・モジュールについて一つとして話を進めたが、実際には複数の入出力部がハードウェア・モジュール上に存在することも多々あり、その数に応じて必要な数だけ入出力部を設けることが可能であることはいうまでもない。

【0147】また、本実施例では処理Cに必要なハードウェア・モジュールとして(C-a)、(C-b)、(C-c)、(C-d)、(C-e)、(C-f)、(C-g)、(C-h)の8つをそれぞれ入出力部を異なった位置に配置したハードウェア・モジュールとして記憶部に格納したが、これは記憶部の容量、考えられるハードウェア・モジュールの種類、あるいは特に使用する可能性が高いハードウェア・モジュールなど、その状況に応じて格納する数、種類などを変えられることはいうまでもない。

【0148】次に、プログラマブル論理回路400上に先に構成されたモジュール(A)に後続して構成される



処理モジュールについて、それぞれ入出力部の異なるモジュールを記憶装置200に蓄積して、最も配線遅延の小さいモジュールを順次選択してプログラマブル論理回路400上に構成する態様についてその処理フローを説明する。

【0149】図10には、この手順をフローチャートの形式で示している。以下、このフローチャートの各ステップについて説明する。

【0150】処理が実行されるアプリケーション・プログラム100は、ハードウェア・モジュールとして実行される3つの処理モジュールA、B、及びCを含み、且つ、 $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$ という順番で各処理モジュールが実行されることとする。

【0151】既に述べたように、ハードウェア・モジュール取得手段300と、実行モジュール決定手段600は、情報処理システム100上で動作するOSにおける各1つの機能、すなわちソフトウェア的に実装されている。したがって、この動作は、アプリケーション・プログラム100の開始にตอบสนองして、OSによって処理が実行される。

【0152】アプリケーション・プログラム100が開始されると、プログラム100内のヘッダ部に記述されているハードウェア・モジュールの識別符号(図5を参照のこと)が読み込まれる(ステップS31)。既に述べたように、識別符号は、各ソフトウェア・モジュールと同じ処理をプログラマブル論理回路16上に再構成する回路情報と、再構成されたハードウェア・モジュールの形状に関する情報を含んでいる。

【0153】ここで、ハードウェア・モジュールの形状は、ハードウェア・モジュールをプログラマブル論理回路16上に再構成して占有するセル領域を完全に包含する矩形で表現することとし、したがって、矩形が持つ水平及び垂直の各方向のセル(PFU)数によって定義されている。

【0154】予定された順序に従って各処理モジュールを逐次実行するという形式で、アプリケーション・プログラム100の実行が進行する。そして、ハードウェア・モジュールによる処理部分に到達すると、必要なハードウェア・モジュールがプログラマブル論理回路16上にコンフィギュレーションされていく。図10の破線で囲んだ部分がこれに相当する。

【0155】本実施例では、アプリケーション・プログラム100は、ハードウェア・モジュールとして実行すべきと決定されている処理モジュールとして、A、B、Cという3つを含み、且つ、処理モジュールの実行順序は $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$ と指定されている。

【0156】まず、ステップS32において、処理Aを実現するハードウェア・モジュール(A)が、プログラマブル論理回路16上にコンフィギュレーションされる。次いで、ステップS33において、処理Aが開始さ

れる。

【0157】上述の処理Aのモジュールのコンフィギュレーションの終了と同時にハードウェア取得手段300は、記憶手段200から、次の処理Bに必要なハードウェア・モジュール(B)の取得を試みる。処理Bに必要なハードウェア・モジュール(B)がプログラマブル論理回路16上にコンフィギュレーションする。ここで、各ハードウェア・モジュールが格納されている記憶手段200には、処理Bに必要なハードウェア・モジュールとして、機能が同じで入出力部の位置が異なっている(B-1)~(B-n)の複数のモジュール群が1つの集合として処理Bに必要なハードウェア・モジュールの対象として格納されている。

【0158】この中でハードウェア・モジュール(A)の入出力部と最短距離で接続可能なハードウェア・モジュールとして(B-i)が選択され(S41)、コンフィギュレーションが実行される(S42)。

【0159】処理Aが終了すると、ハードウェア・モジュール(B)のコンフィギュレーションが終了したかどうかを確認して(S34)、終了している場合は処理Bを実行する(S35)。

【0160】ハードウェア・モジュール(B-i)のコンフィギュレーションが終了すると同時にハードウェア取得手段300は、記憶手段200から、次の処理Cに必要なハードウェア・モジュール(C)の取得を試みて、処理Cに必要なハードウェア・モジュール(C)をコンフィギュレーションする。ここで、各ハードウェア・モジュールが格納されている記憶手段200には、処理Cに必要なハードウェア・モジュールとして、機能が同じで入出力部の位置が異なっている(C-1)~(C-n)の複数種類を1つの集合として処理Cに必要なハードウェア・モジュールの対象として格納している。

【0161】この中でハードウェア・モジュール(A)及びハードウェア・モジュール(B)の入出力部と最短距離で接続可能なハードウェア・モジュールとして(C-i)が選択され(S43)コンフィギュレーションが実行される(S44)。

【0162】処理Bが終了すると、処理Cに必要なハードウェア・モジュール(C)がコンフィギュレーションされたかどうかを確認して(S36)、コンフィギュレーションされていれば処理Cを実行する(S37)。処理Cが終了すると、この処理が1回目の処理であるかどうかを確認し(S38)、1回目であれば、処理Aの実行(S33)に戻り、そうでなければハードウェア・モジュールでの処理を終了する。

【0163】この実施態様では $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$ の処理態様について説明したが、さらに $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ の態様、あるいは、 $A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow D$ のような態様と様々な処理態様である場合にも、それぞれプログラマブル論理回路16上に先に構成されたハ



ードウェア・モジュールに遅れて構成されるハードウェア・モジュールの各々について、先に構成されたモジュールとの入出力位置を最短とするモジュールを選択する構成とすることによって、前述の図9で説明した信号伝達に要する時間、例えばTAB、TCA等、より一般的にはモジュールXとモジュールYとの間の信号伝達に要する時間TXYを短縮することが可能となり、その結果として全体の処理時間を短縮化することができる、優れた情報処理システムを提供することができる。

【0164】以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【0165】

【発明の効果】以上詳記したように、本発明によれば、論理回路を記述した回路情報に従って所望の論理を再構成可能なプログラマブル論理回路を含んだタイプの、優れた情報処理システムを提供することができる。

【0166】また、本発明によれば、プログラム言語で処理を記述された複数の処理モジュールで構成される一連の処理の少なくとも一部を部分書き換え可能なプログラマブル論理回路上で再構成したハードウェア回路に置き換えて実行するタイプの、優れた情報処理システムを提供することができる。

【0167】また、本発明によれば、各処理モジュールをソフトウェア・モジュール又はハードウェア・モジュールのいずれで実行すべきかを決定して処理するタイプの、優れた情報処理システムを提供することができる。

【0168】また、本発明によれば、プログラム論理回路の異なるモジュール間のデータ入出力に要する時間を減少して処理モジュール全体の処理時間を短縮化することができる、優れた情報処理システムを提供することができる。

【0169】また、本発明によれば、プログラマブル・デバイス上に構成するハードウェア・モジュールを機能が同じで入出力部の位置が異なる態様の複数モジュールを選択して構成可能としたので、モジュール間の配線遅延を減少することができ、また、その結果として全体の処理時間を短縮化することができ、優れた情報処理システムを提供することができる。

【図面の簡単な説明】

【図1】本発明を実現するのに適した情報処理システム10のハードウェア構成を模式的に示した図である。

【図2】プログラマブル論理回路16の構造を論理的に図解したものである。

【図3】プログラマブル論理回路16の物理構造を模式的に示した図である。

【図4】情報処理システム10が複数の処理モジュールからなる一連の処理を実行する様子を模式的に示した図である（但し、一連の処理は、プログラム言語で記述された各処理モジュールで構成されるプログラムの形態で、システム10に供給されているものとする）。

【図5】一連の処理を構成するアプリケーション・プログラムの構成を模式的に示した図である。

【図6】ハードウェア・モジュールの入出力部位置関係を模式的に示した図である。

【図7】ハードウェア・モジュールによって処理モジュールを実行する手順を示したフローチャート（その1）である。

【図8】入出力部位置が異なるが同じ処理を実行する複数のハードウェア・モジュールの中から最適なものを選択してプログラマブル論理回路16上に再構成した本発明の構成を従来例と対比して示した図である。

【図9】本実施例に従って処理A→処理B→処理C→処理A→処理B→処理Cの順で各ハードウェア・モジュールが実行される処理フローを、従来例と比較しながら、時系列的に表示した図である。

【図10】ハードウェア・モジュールによって処理モジュールを実行する手順を示したフローチャート（その2）である。

【図11】特開平9-74556号公報に開示された画像再生装置の構成を模式的に示した図である。

【図12】特開平6-301522号公報に開示された計算機システムの構成を模式的に示した図である。

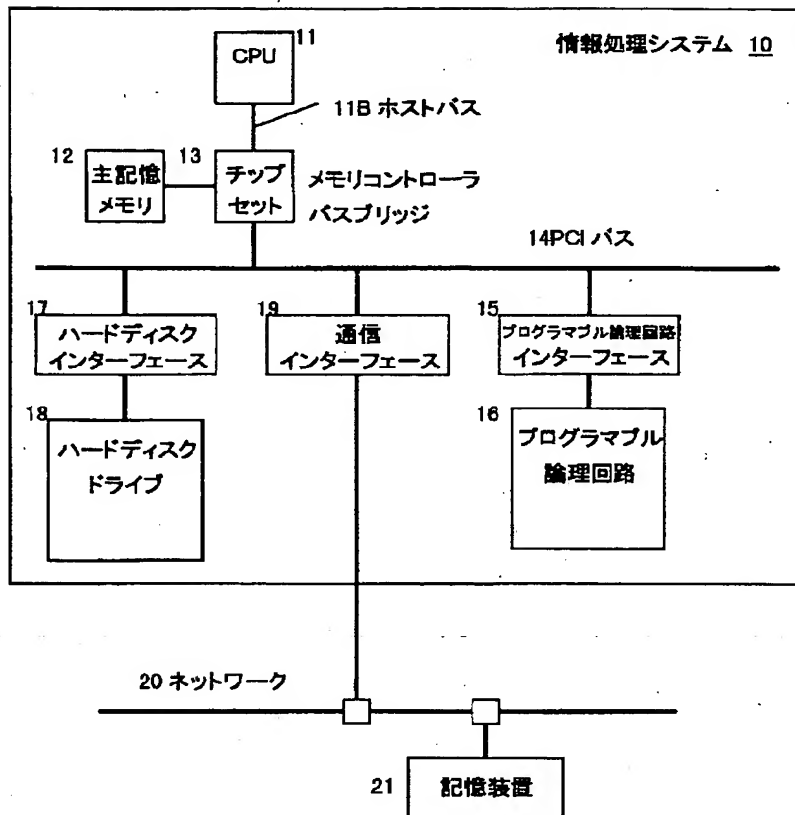
【符号の説明】

- 10 情報処理システム
- 11 CPU
- 11B ホストバス
- 12 主記憶メモリ
- 13 チップセット
- 14 PCIバス
- 15 プログラマブル論理回路インターフェース
- 16 プログラマブル論理回路
- 17 ハードディスクインターフェース
- 18 ハードディスクドライブ
- 19 通信インターフェース
- 20 ネットワーク
- 21 記憶装置
- 100 プログラム
- 161 論理セル
- 162 配線領域
- 163 入出力端子
- 200 記憶装置
- 300 ハードウェア・モジュール取得手段
- 310 取得要求送出手段
- 320 取得再構成手段
- 400 プログラマブル論理回路

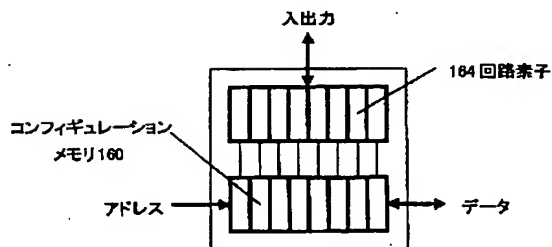
500 CPU  
600 実行モジュール決定手段

800 識別符号記録手段

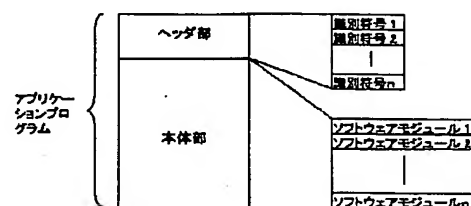
【図1】



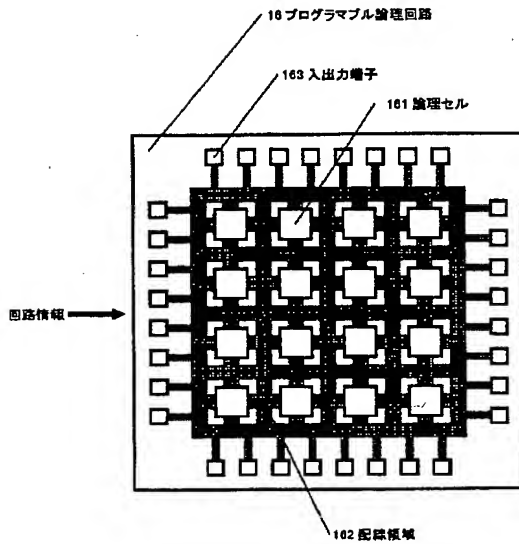
【図2】



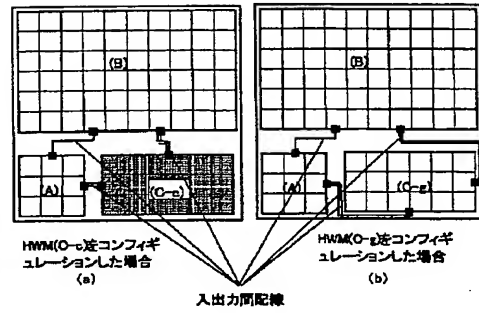
【図5】



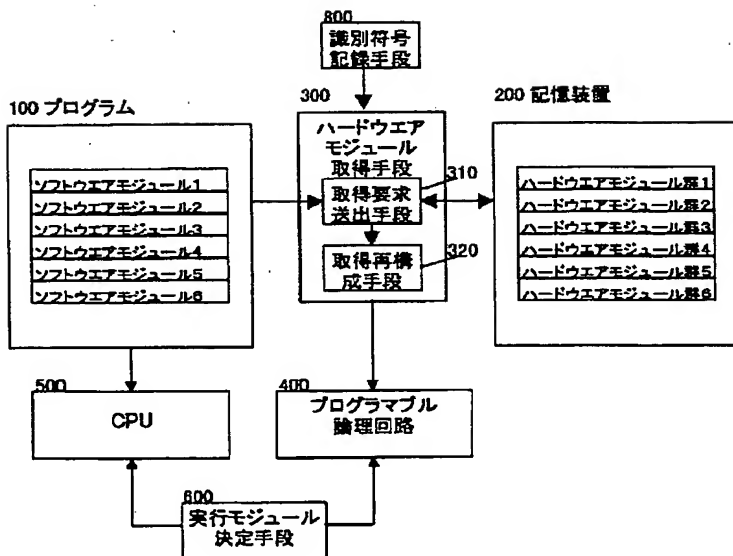
【図3】



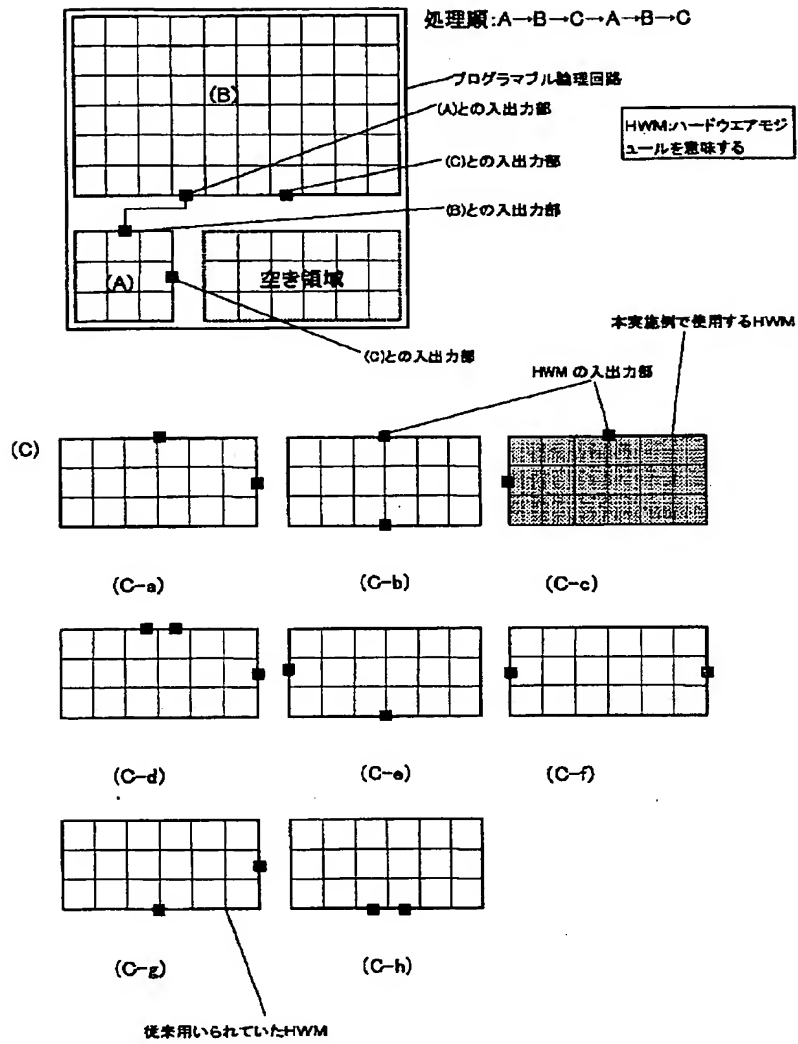
【図8】



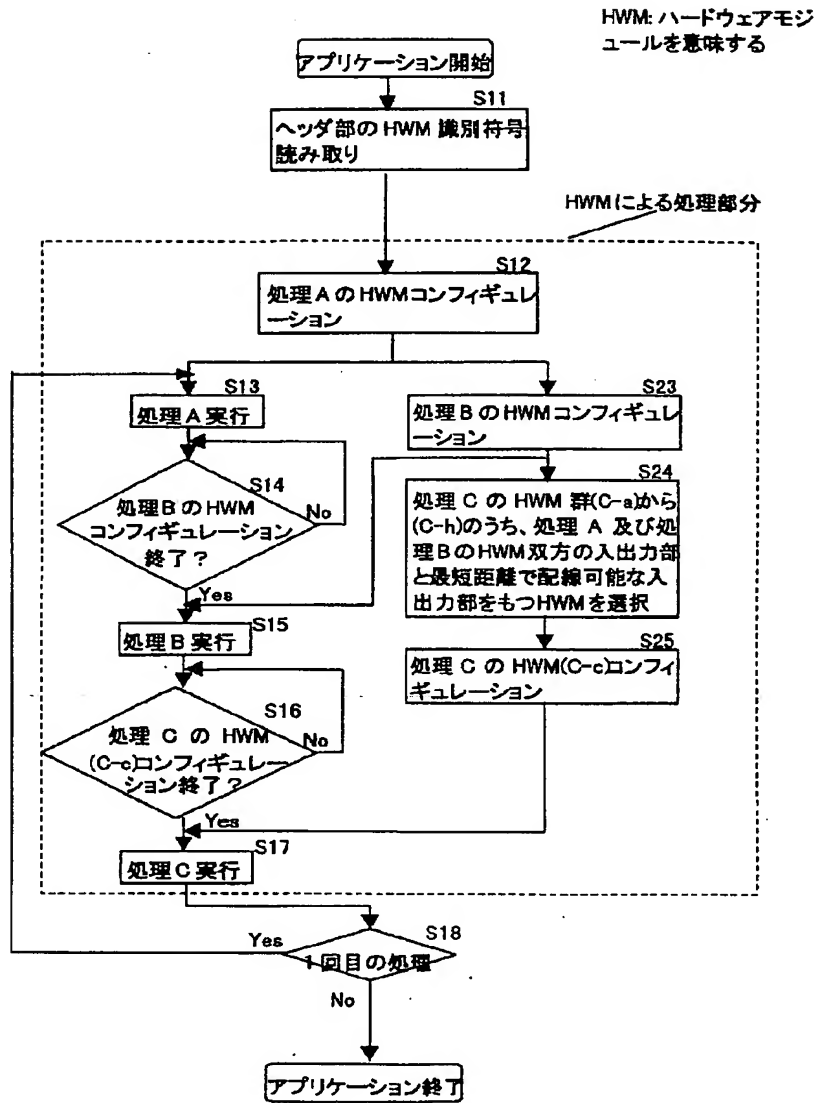
【図4】



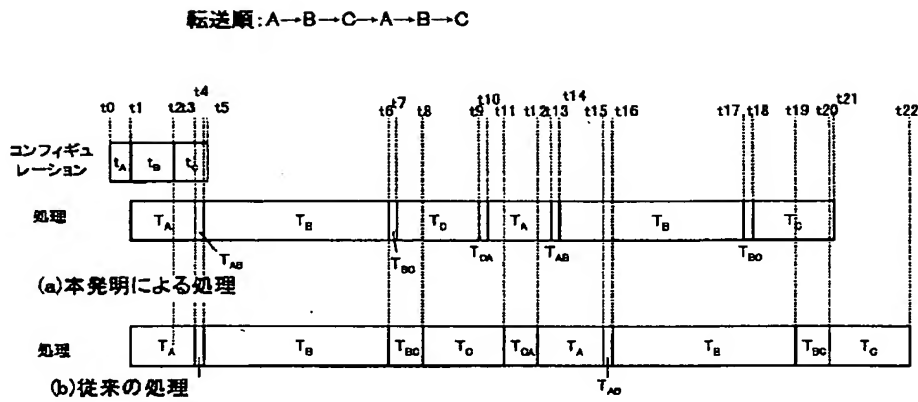
【図6】



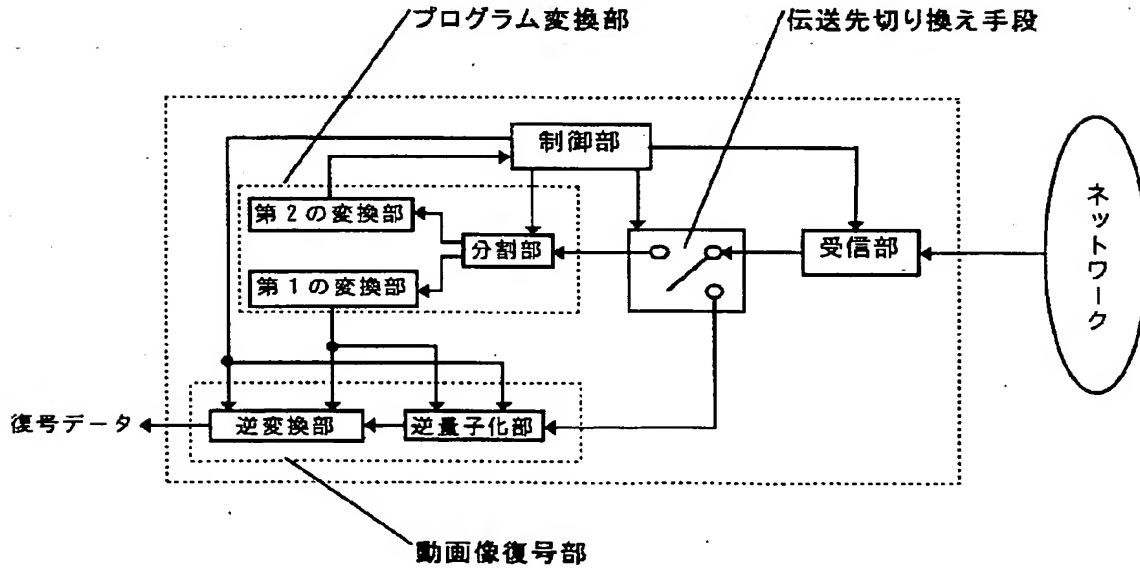
【図7】



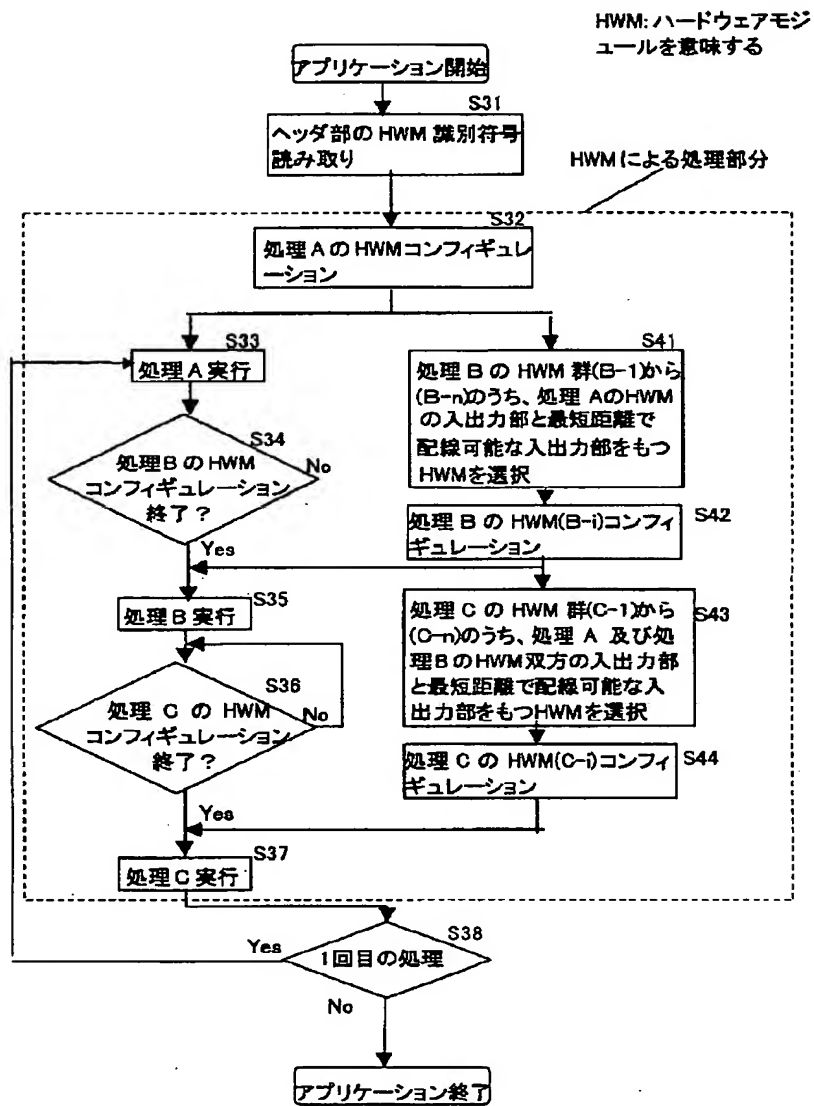
【図9】



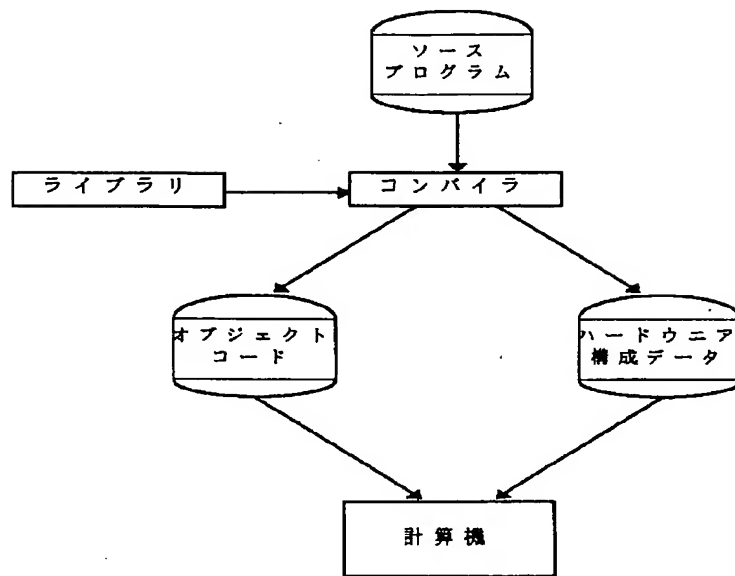
【図11】



【図10】



【図12】





**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**